Task Selection and Planning in Human-Robot Collaborative Processes: To be a Leader or a Follower?

Ali Noormohammadi-Asl[1*], Ali Ayub[1], Stephen L. Smith[1], and Kerstin Dautenhahn[1]

***Abstract*— Recent advances in collaborative robots have provided an opportunity for the close collaboration of humans and robots in a shared workspace. To exploit this collaboration, robots need to plan for optimal team performance while considering human presence and preference. This paper studies the problem of task selection and planning in a collaborative, simulated scenario. In contrast to existing approaches, which mainly involve assigning tasks to agents by a task allocation unit and informing them through a communication interface, we give the human and robot the agency to be the leader or follower. This allows them to select their own tasks or even assign tasks to each other. We propose a task selection and planning algorithm that enables the robot to consider the human's preference to lead, as well as the team and the human's performance, and adapts itself accordingly by taking or giving the lead. The effectiveness of this algorithm has been validated through a simulation study with different combinations of human accuracy levels and preferences for leading.**

I. INTRODUCTION

The emergence of collaborative robots (cobots) has freed robots from their cages and enabled them to collaborate with humans safely in shared workspaces. Through this collaboration, human-robot complementary skills can be leveraged to achieve more effective and efficient performance. While robots are known to be fast, powerful, accurate, precise, and able to repeat actions without tiring, due to their limited cognitive capabilities, it is still challenging for them to work autonomously in unstructured and changing environments with flexible tasks [1]. This is where humans' superior cognitive capabilities can help the human-robot team adapt and cope with these changes and uncertainties. The presence of the human requires robots to come up with more complicated plans that consider their human teammates and their resulting non-determinism [2].

Unlike some human-centered designs in which the robot has to adapt itself one-sidedly to the human's needs and preferences, in industrial settings, the performance of the system must also be at a high level to economically justify the deployment of cobots. In other words, in addition to the human-related factors (e.g, their mental and physical comfort), other system performance metrics such as completion time, energy, and accuracy must also be taken into account in the design and planning of the collaboration. Additionally, our previous user studies showed no significant difference in participants' perception of the robot and collaboration between the case where the robot prioritized the human's objectives and the one where the robot took a balancing strategy between its and the human's objectives [3]. In other words, a plan that considers both the human preferences (goals) and the team objectives can enhance the performance of the team without negatively impacting the human perception of the robot.

This research was undertaken, in part, thanks to funding from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canada 150 Research Chairs Program.

[1]University of Waterloo, Waterloo, Ontario, Canada
[*]a.noormohammadi@uwaterloo.ca

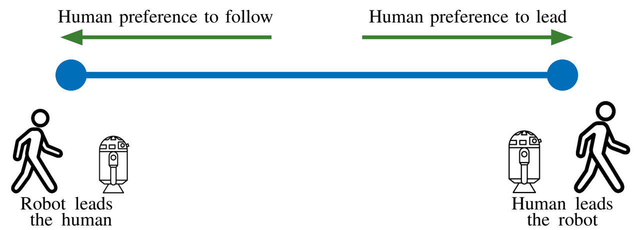


Fig. 1. The spectrum of the human's leading or following role in a human-robot collaborative task

In this paper, we study the task planning problem in a human-robot collaborative system to find a schedule for the robot that optimizes the collaboration objectives, including completion time, accuracy, and the human's preferences. Most studies in the literature assume that there is a role allocation unit (e.g., robot) which takes the leading role and assigns tasks to corresponding agents. For instance, in the robotic picking assistive system used by DHL [4], the human workers are required to follow the robots to the picking location. In [5], the authors compare the team performance in a warehouse order picking scenario where the human either leads or follows the robot. The human agents may, however, deviate from the plan due to different reasons, such as different preferences, fatigue, changes in the environment, or unexpected demands. In the scenario we designed, instead of being on opposite ends of the spectrum, namely the human either leading or following the robot (Fig. 1), we assume that each agent has the agency to choose their own tasks and to assign tasks to the other. Thus, the robot planner needs to select tasks while considering team efficiency and the human's preference between taking the leading or following role. In addition, the robot should monitor the human's performance and take the leading role if the human is not efficient and precise.

A. Contributions

Four contributions of our work stand out: 1) We design a simulated scenario in which both the robot and the human have the agency to select their own tasks or even assign tasks to each other. 2) We propose a task selection algorithm that enables the robot to adapt its leading role online based on the human's performance and preference to follow or lead. 3) We provide a dynamic task planning approach to update the task and plan and to fix the human's errors. 4) Finally, we validate our method through simulation for different combinations of human accuracy and preference to lead.

B. Related work

In the literature on human-robot collaboration, the two problems of task scheduling (planning) and robot adaptation to humans have been studied extensively. The former mainly focuses on creating optimal plans for assigning the human-robot team and exploiting their capabilities, and the latter attempts to make robots adjust their actions to humans' preferences and actions. Although our work does not exclusively fall into only one of these two problems, the related literature provides crucial insight into approaches and tools that we can adopt to bridge the gap between the two.

1) Task scheduling: The problem of task scheduling for industrial robots, without any human teammate present, has been investigated in many studies [6]–[8]. The goal of these systems is to minimize processing costs and time. However, it is not possible to apply those methods directly to a team consisting of humans and robots. Instead, offline scheduling that considers the different capabilities and characteristics of both human and robot can be employed, as in [9], [10]. The uncertainty due to the presence of the human, however, necessitates dynamic and online task scheduling [11]–[13]. In [14], [15], the authors propose adaptive task scheduling algorithms that monitor the human's activities and adapt the robot's plan based on their capabilities and quality of work.

Human preference is an essential factor that has been overlooked in many task scheduling studies. The authors in [16] studied scenarios in which tasks were allocated by either the manager, the robot, or the participants. In [17], the human preference to do certain types of tasks is also considered in the offline planning. The human-subject experiments in both studies have shown that participants' agency significantly improves their perception of the collaboration.

2) Adaptation: Robot adaptation to humans is important for smooth and effective human-robot collaboration. A basic idea in adapting robots to humans is employing experts and developers to manually teach the robot how to collaborate to finish a task while complying with human preferences [18], [19]. Supervised learning-based algorithms have also been employed to learn human preferences online [20]–[22]. In these types of algorithms, the decision factors (features) that impact the human's behavior are identified, and data is collected on human behavior. The obtained data is enriched by means of expert annotation or by surveying participants.

Unsupervised learning-based methods enable machines or robots to learn human preferences after observing their behavior and actions [23]. This approach has been studied for modeling human adaptive behaviors [24], predicting human reaching to avoid interference [23], and learning the human model from joint-action demonstrations [25]. Some work also exploits both learning methods and experts' knowledge [26]. In [24], the authors consider a mutual human-robot adaptation rather than one-way adaptation of the robot to the human, enabling the robot to guide an adaptable human.

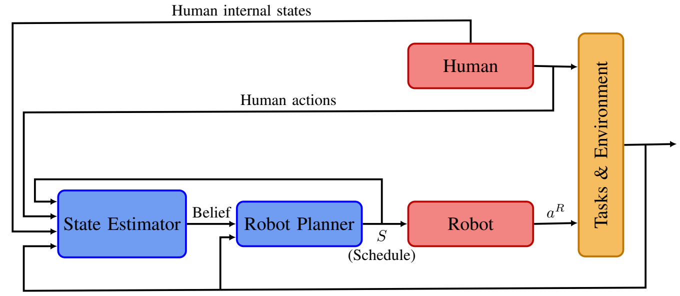


Fig. 2. Task selection and planning architecture

II. PROBLEM STATEMENT

The collaborative task that we consider in this paper involves two agents: a human and a robot. These two agents need to collaborate to accomplish a set of precedence-constrained tasks $\tau = \{\tau_1, \tau_2, \dots, \tau_n\}$. The required times to complete a task τ_i are t_i^h and t_i^r for human and robot, respectively. Due to uncertainties of the agents and environment, the real execution time of a task may be different from the nominal required time. In each decision step, an agent selects a feasible task to perform itself and can also assign a set of feasible tasks to the other agent. The human's and the robot's agency to choose their actions or assign tasks to each other differentiate this problem from typical task allocation and scheduling problems where agents are assigned tasks and are informed what and when they must carry them out. This problem poses some challenges for robot planning:

- The robot must estimate the human preference to be the leader (i.e., the robot follows them) or the follower (i.e., they follow the robot) and select actions accordingly.
- The robot must monitor the human actions' effect on team performance and adapt its leading role accordingly.
- The robot has to select actions that minimize collaboration cost. Here, we only consider the completion time.
- The robot has to detect and correct human errors.

A. Planning Architecture

The strategy that we employ for the robot's task selection and planning consists a state estimator and a planner (Fig. 2).

State estimator: During collaboration, the robot considers the human's actions and estimates their preference for leading or following. Furthermore, it needs to monitor the human's performance and estimate their accuracy level. Since these states cannot be measured directly, the robot has to estimate them through the history of the interaction. To estimate the human's states, the state observer takes the following inputs: the history of the human's actions, the robot's previous beliefs and schedule, the human's internal states (e.g., speed and fatigue), and the tasks' states.

Planner: The planning unit comprises two phases: task selection and task scheduling. In each decision step, if

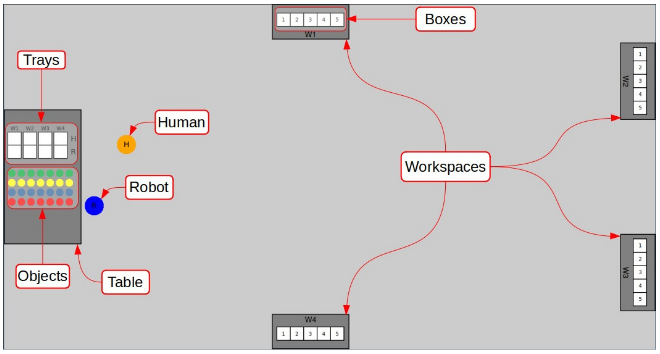
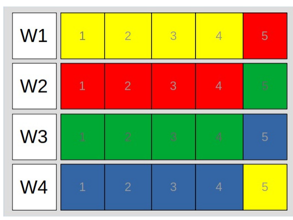
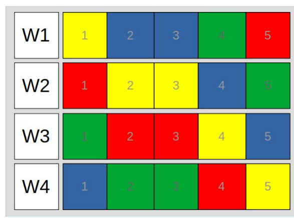


Fig. 3. Simulation environment: In a $17 \times 9m$ room, there are four workspaces (W1-W4) with five boxes on each. There are 28 objects of four different colors (7 of each) on the table. Additionally, there is a tray for the human and one for the robot. Each tray has four compartments, labeled W1-W4. A video of the simulation is available at *https://youtu.be/gEzNoiiGP-w*



(a) Pattern 1: An easy one to memorize (Difficulty: Easy)



(b) Pattern 2 (Difficulty: Medium)

Fig. 4. Humans are shown patterns with different levels of memorization difficulty.

needed, the robot first does the task selection and then performs the task scheduling to select its next action.

B. A Sample Scenario

Here, we provide a possible real-world collaborative scenario of the explained problem to help readers better understand the objectives of this study, which are encapsulated in Fig 1. The simulation results in Section IV are based on this experiment. In this scenario (Fig. 3), the human and robot have to collaborate to fill a certain number of boxes in each workspace (W1-W4) with the correct objects.

As shown in Fig. 3, there are four workspaces, W1-W4, with five empty numbered boxes on each. In the envisaged future human-cobot experiment, at the beginning of the experiment, a pattern (e.g., Fig. 4a or 4b) is displayed to the human-robot team depicting how the boxes must be filled. The pattern is shown for a short amount of time, and the human needs to memorize it. Depending on the pattern's difficulty level, the human may make errors or need the robot's help to complete the tasks. The boxes on each workspace are numbered and objects have to be placed in the correct order. For instance, in a real-world workspace, agents cannot place an object into box 3 before filling boxes 1 and 2. On the large table, there are two trays, which are empty at the beginning of the experiment. Each tray has four compartments, labeled W1-W4. The upper one in Fig. 3, labeled *'H'*, is where the robot can assign objects to the human. Similarly, the human can assign objects to the robot and put them on the tray, labeled *'R'*. For example, when the robot places an object in the human's tray labelled 'W1', the next object that the human has to put in workspace 1 (W1) is the one that is in the tray. In addition, when one of the compartments in a tray is full, the other agent is not allowed to put another object in the same-labeled compartment of the other tray. For instance, when the robot has already placed an object in one of the human tray's compartments labelled 'W2', the human cannot place any object in the compartment with the same label, 'W2', on the robot's tray.

We assume that both robot and human decide when they will approach the table. In addition, a safety zone is imposed around the table, which an agent cannot enter when the other agent is inside (to account for e.g. safety considerations). Upon approaching the table, the human has to take one or more feasible action from the following set of actions:

1) pick up an object from the table or the human's tray and place it in the workspace.
2) place an object in the robot's tray.
3) pick up an object from the robot's tray and place it in the workspace.

In each step, the human needs to take one of Actions 1 or 3 to exit the table area so that the robot can enter it. The human can select Action 3 only when Actions 1 and 2 are not feasible.

The robot is also responsible for a set of actions, similar to those of the human. The robot also needs to correct possible human errors. Thus, the robot's set of actions are as follows:

1) pick up an object from the table or the robot's tray and place it in the workspace
2) place an object in the human's tray
3) bring back a misplaced object from the workspace
4) bring back a misplaced object from the robot's tray
5) pick up an object from the human's tray and place in the workspace

Similar to the human, the robot also needs to take one of Actions 1, 3, or 5 to exit the table area. Action 5 can be chosen only when Actions 1-4 are not feasible. The collaboration objective is accomplished when all objects are put in the correct boxes.

III. Planning Strategy

At each decision (action) step, the robot needs to develop a plan for the one-to-one assignment of agents to tasks, along with a time scheduling to determine when tasks need to be done. To do so, the robot needs to consider two factors: 1) the collaboration time, and 2) the human performance and preference to lead. Task allocation and scheduling problems can usually be modeled as mixed linear integer programs (MILP). However, the complexity of MILP-based solutions makes them computationally intractable. Decomposing task allocation and task scheduling is a promising approach to deal with this complexity [27]. Creating and solving a single optimization problem becomes more challenging and arduous due to changing and uncertain factors regarding the human's behaviour and intentions. Thus, we split the problem into two subproblems: task allocation and task scheduling.

In the first phase, given the set of agents $A = \{\textit{human}, \textit{robot}\}$, set of tasks $\tau = \{\tau_1, \tau_2, \ldots, \tau_n\}$ and their corresponding costs of assigning them to the human and robot $C_{\tau_i}(a), a \in A$, the robot first solves for an optimal task allocation. Then, if needed, a new set of tasks τ_{new}, including actions required to assign tasks to the human, is created. The task scheduler uses the obtained optimal task allocation and τ_{new} and solves for an optimal task schedule. If the solution from the task allocation phase does not lead to a feasible solution in the task scheduling phase, the first phase needs to be redone to obtain a new allocation.

As shown in (1), the goal of task allocation is to minimise the maximum cost of assigning the tasks, between the human and robot.

$$\mathbf{X}^* = \min_{\{\mathbf{X}\}} \max_{A} \mathbb{E}\left[\sum_{\tau_i \in \tau, a \in A} X^a_{\tau_i} C_{\tau_i}(a) \right] \tag{1}$$

subject to

$$\sum_{a \in A} X^a_{\tau_i} = 1, \quad \forall \tau_i \in \tau \tag{2}$$

$$\textit{problem-dependent constraints}. \tag{3}$$

In this optimization problem, $X^a_{\tau_i} \in \{0, 1\}$ is a binary decision variable and equals 1 when task τ_i is assigned to agent $a \in A$. We also define $\mathbf{X} = \{X^a_{\tau_i} \mid \tau_i \in \tau, a \in A\}$. Function C_{τ_i} is the cost incurred by assigning the task to the robot or the human while considering human's performance P_e, and preference to follow the robot P_f. Related to P_f, function C_{τ_i} incurs a higher cost for assigning the tasks to the human who prefers to lead, and a lower cost for assigning tasks to a faulty human (high P_e), because assigning tasks by the robot restricts the human's agency and avoids human errors. Equation (2) ensures that each task is assigned only to the human or the robot. There are additional problem dependent constraints, which can be added to (3). For instance, in the scenario explained, the robot needs to assign tasks so that the robot can take one of either Actions 1, 3, or 5.

Having accomplished the optimal task allocation and updating the task, the robot needs to find an optimal schedule to determine what and when tasks are needed to be performed. We define the decision variables $s_{\tau_i}, i = 1, \ldots, n$ as the start time of tasks in τ. The variable f_{τ_i} is also the finish time of task τ_i. To consider task precedence we adopt a binary function $P(\tau_i, \tau_j)$ which equals 1 if τ_i has to be finished before τ_j. In addition, $Q(\tau_i, \tau_j)$ is a binary decision variable, and $Q(\tau_i, \tau_j) = 1$ if τ_i and τ_j are assigned to the same agent and τ_i comes before τ_j.

Here, as we only consider the collaboration time, the task scheduling problem can be written as minimising the overall processing time:

$$\min \max_{\tau_i \in \tau_{new}} f_{\tau_i} \tag{4}$$

subject to

$$P(\tau_i, \tau_j) . f_{\tau_i} \leq s_{\tau_j}, \qquad \forall \tau_i, \tau_j \in \tau_{new} \tag{5}$$

$$Q(\tau_i, \tau_j) . f_{\tau_i} \leq s_{\tau_j}, \qquad \forall \tau_i, \tau_j \in \tau_{new} \tag{6}$$

$$f_{\tau_i} = s_{\tau_i} + d_{\tau_i}, \qquad \forall \tau_i \in \tau_{new} \tag{7}$$

$$\textit{problem-dependent constraints}. \tag{8}$$

Algorithm 1: Task selection and planning

input : Precedence-constrained task, τ

1 $P_f \leftarrow$ Initial belief about human preference to follow
2 $P_e \leftarrow$ Initial belief about human performance
3 **while** *Tasks are not finished* **do**
4   Monitor the human's actions
5   Detect the human's wrong actions
6   Update task, τ
7   Update P_f and P_e
8   **if** *new schedule is needed* **then**
9   **while** *schedule, $S^\star$, is not found* **do**
10             $X^\star \leftarrow$ `TaskSelection(`τ, P_f, P_e`)`
11             $\tau_{new} \leftarrow$ `CreatTempTask(`τ, $X^\star$`)`
12             $S^\star \leftarrow$ `TaskSchedule(`τ_{new}, $X^\star$`)`
13     $a^R \leftarrow$ `GetAction(`$S^\star$`)`
14   `ApplyAction(`a^R`)`

Inequality (5) enforces the precedence constraints. Inequality (6) ensures that no agent can do more than one task at any one time. In (7), f_{τ_i} depends on the time the assigned agent requires to finish it, d_{τ_i}. Similarly, constraints of this optimization problem are problem-dependent, and the aforementioned constraints can be modified, and other ones can be added (8). For example, in the scenario designed, the task scheduling has to consider that if the robot needs to take any of Actions 2 or 4, it must be done before Actions 1, 3, or 5.

Optimization problems (1) and (4), based on the problem-dependent constraints, can be expressed as mixed-integer linear or nonlinear programs. In our experiment, however, they can be rewritten in a linear form and solved by the robot using solvers such as Gurobi or CPLEX.

A. Algorithm

The task selection and planning procedure proposed in this paper is explained in Algorithm 1. The robot starts with an initial belief about the human's performance level and preference to follow the robot (Lines 1-2). During the task, the robot needs to monitor and record the human's actions and mistakes (Lines 4-5). The robot also has to update the task and precedence constraints based on the finished tasks and the human's mistakes that are needed to be corrected (Line 6). Then, according to the structure shown in Fig. 2, P_f and P_e needs to be updated (Line 7). Next, if a new plan is required, the robot must solve for a new one, as explained before (Line 9-12). Finally, the robot performs its action, a^R based on the obtained schedule (Lines 13-14).

IV. Simulation Experiments

The experiments in this section are based on the scenario explained in Section II and Fig. 3.

Collaboration task: The tasks and precedence constraints are encoded by a directed acyclic graph, called a task graph, where vertices are tasks and edges represent precedence

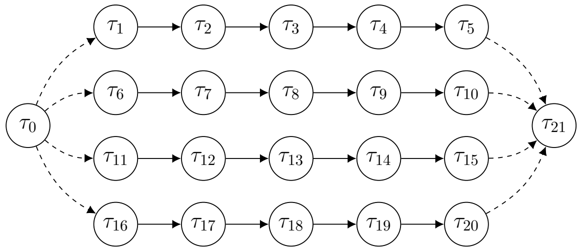


Fig. 5. Task graph of the experiment

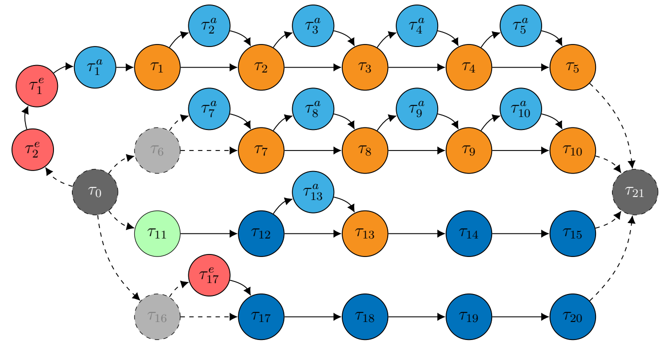


Fig. 6. Temporary task graph of the experiment after task allocation. Blue: Robot's tasks, Orange: Human's tasks, Cyan: Assigning tasks to the human, Red: Correcting human errors, Green: Already assigned tasks, Gray: Finished tasks

constraints between them. The initial task graph of the experiment, with two dummy nodes as starting (T_0) and finishing (T_{21}) points, is depicted in Fig. 5. In this experiment, we assume that the human's speed does not change during the experiment and is higher than that of the robot. Thus, the nominal required processing times for tasks are fixed. However, in a future real-world scenario, this assumption can be easily relaxed if the robot can measure the human's speed and update processing times accordingly.

Human following preference: In this study, with a single scalar random variable p_f, we capture the human preference for following the robot. In addition, we assume a tuple of possible discrete values for p_f, given by $V = \big(0.0, 0.2, 0.4, 0.6, 0.8, 1\big)$. The values $p_f = 0$ and $p_f = 1$ respectively mean that the human very likely prefers to lead and follow the robot. At the beginning of the experiment, the robot assumes that the human will follow it, and sets its initial belief about that as follows:

$$P[p_f = v_i] = b(i; n = 5, p = 0.8),\ V = (v_0, \dots, v_5),$$

where $b(i; n, p)$ is a binomial distribution.

Human performance: In this experiment, we only consider the probability that the human may make a mistake and put wrong objects in the workspaces and the robot's tray. A single scalar random variable p_e, with a tuple of discrete values $W = \{0, 0.1, 0.2, \dots, 1\}$, is adopted to capture the probability of human error. Humans with $p_e = 1$ make too many errors, and in contrast, $p_e = 0$ indicates they are accurate. The robot first assumes that the human is almost accurate, and during the collaboration updates its initial belief, which is a binomial distribution

$$P[p_e = w_i] = b(i; n = 10, p = 0.2),\ W = (w_0, \dots, w_{10}).$$

Task allocation: The task allocation problem is modeled as (1), where

$$C_{\tau_i}(a) = \begin{cases} t_i^h p_f + c_f(1 - p_f) & a = \mathit{Human} \\ t_i^r + p_e c_e + c_h x_{\tau_i}^{\mathit{human}} & a = \mathit{Robot} \end{cases}. \tag{9}$$

c_f is the penalty value for assigning the task to the human, who prefers to lead, and c_e is the penalty imposed for not assigning tasks to the human making mistakes. By assigning task to the human, the robot informs the human about the next object that has to be put in a workspace. This restricts the human's agency and avoid the human making mistakes. In addition, penalty c_h is incurred for assigning the robot a task whose corresponding object has already been placed on the human's tray ($x_{\tau_i}^{\mathit{human}} = 1$). Fig. 6 shows an example of task allocation and the temporary task graph. In this case, the human has made a mistake in tasks 1, 2, and 17. Thus, the robot needs to fix them (e.g., T_2^a). tasks 6 and 16 are finished, and the robot has placed an object in the human's tray for task 11. The robot has assigned tasks 1, 2, 3, 4, 5, 7, 8, 9, 10, and 13 to the human and needs to put the tasks' corresponding objects on the human's tray (e.g., T_7^a). The robot has also selected tasks 12, 14, 15, 17, 18, 19, and 20 to do by itself.

Solving optimization problems: The task allocation and task scheduling problems have been rewritten as mixed integer linear program. Both problems are considered as NP-hard optimization problems. We implemented the simulation experiments on a computer with an Intel Core i7-8700 CPU 3.20GHz × 6-cores CPU and 16 GB RAM, on Ubuntu 20.04. We used the GUROBI mathematical optimization solver and set a time limit to force the solver to exit with the current solution if it still searches for other solutions. To accelerate the computational time, we have employed a warm-start – providing the solver a (partially) valid initial solution – that uses the current step's solution in the subsequent step. Hence, for the first step, we can solve the problem offline.

Updating p_e and p_f: We used the method proposed in [24] and created some models to update the robot's belief about the human's performance level and preference to follow the robot. Details of the models and the method is explained in the Appendix.

Robot action: The robot chooses its action based on the created plan. Potential field based motion planning is employed for the robot to avoid it colliding with the human.

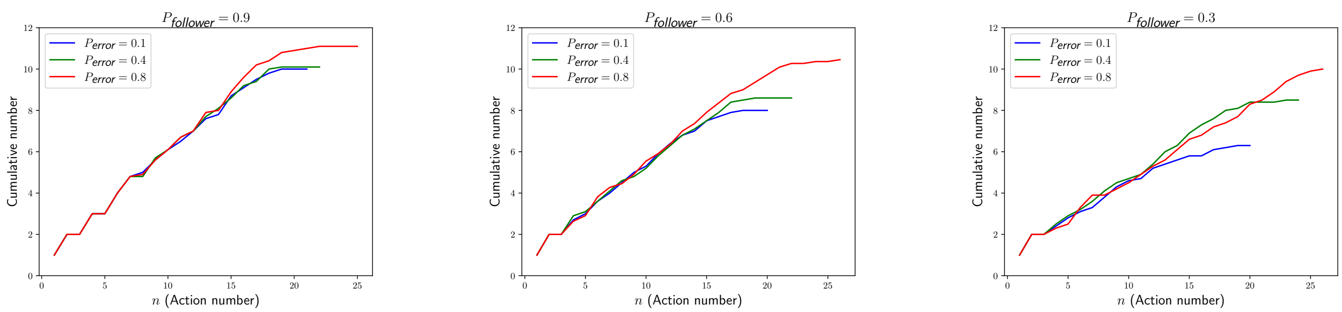
A. Simulation Results

This study conducted simulation experiments to evaluate how the proposed algorithm deals with different combinations of human preferences and accuracy levels. Nevertheless, a real-world human-subject study is necessary to fully assess the algorithm's efficiency, and this will be the aim of our ongoing studies. For this simulation, we created a simple human model which selects actions from the action

TABLE I

SIMULATION RESULTS: THE MEAN VALUE OF RESULT AFTER TEN REPETITIONS FOR EACH CASE.

	Human parameters		Wrong actions		Assigned tasks		Total time (s)		Travel distance (m)		Total actions		Idle time (s)
	P_{follow}	P_{error}	n_h^{wrong}	t_r^{wrong}	n_h^{assign}	n_r^{assign}	t_h^T	t_r^R	d_h^T	d_r^R	n_h^T	n_r^T	t_r^{idle}
Considering human preference and accuracy	0.9	0.1	0.1	1.0	0.1	10.0	133.2	141.3	22310.9	17681.5	11.8	19.3	11.9
	0.9	0.4	1.1	15.5	0.1	10.1	142.8	155.4	24107.8	19415.4	12.7	20.4	11.5
	0.9	0.8	1.9	29.4	0	11.1	154.2	169.8	25938.8	21181.7	13.6	21.9	13.2
	0.6	0.1	0.5	7.7	1.7	8	140.8	146.9	23378.9	18151.1	13.5	17.7	14.0
	0.6	0.4	2.4	23.6	1.6	8.6	146.2	156.7	24771.0	19576.5	14.9	19.7	11.5
	0.6	0.8	4.3	46.9	1.1	10.3	166.3	184.5	28762.6	23921.1	15.8	23.9	20.5
	0.3	0.1	0.9	9.7	4.7	6.3	142.1	149.2	22687.8	17803.4	15.7	17.2	19.5
	0.3	0.4	3.1	22.1	5.5	8.5	152.6	160.5	25351.0	19824.6	18	21.5	15.2
	0.3	0.8	8.5	46.3	6.6	10.1	161.7	172.1	26837.0	20673.6	21.4	27.2	16.1
$p_e = 0$, $p_f = 1$	0.3	0.1	1.4	9.4	6.3	9.3	134.4	140.8	22664.6	17978.0	19.5	18.5	11.0
	0.3	0.4	5	23.2	7.4	10.4	145.6	149.3	24793.9	19073.3	21.3	23.6	10.5
	0.3	0.8	11.4	76.8	7.5	11.7	191.8	200.1	32841.5	25888.7	25.8	30.8	12.9



(a) Strong preference to follow the robot (b) Moderate preference to follow the robot (c) Slight preference to follow the robot

Fig. 7. Cumulative number of assigned tasks to the human by the robot

set discussed in Section II, based on two different parameters: the preference to follow the robot P_{follow} and the probability of forgetting the pattern and choosing wrong objects P_{error}. For the simulation, we consider three different cases of preferences to follow: strong ($P_{follow} = 0.9$), moderate ($P_{follow} = 0.6$), and slight ($P_{follow} = 0.3$). We also take into account three different levels of accuracy: high ($P_{error} = 0.1$), moderate ($P_{error} = 0.4$), and low ($P_{error} = 0.8$).

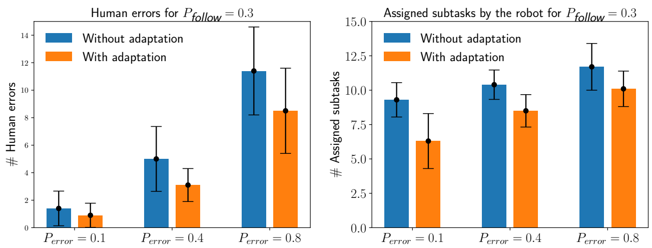
Table I, shows the results of ten simulations for each combination of the human's preferences and accuracy levels. To assess how considering the human's preference and accuracy level affects collaboration performance, we repeated the simulation for three different cases in which the robot takes only the completion time into account and not the human's accuracy or preference for leading (i.e., $p_e = 0, p_f = 1$).

Strong preference to follow the robot ($\mathbf{P_{follow} = 0.9}$): When the human prefers to follow, as the number of assigned tasks indicates n_r^{assign}, the robot takes the leading role and guides the team. In this case, even when the human's accuracy would otherwise be low, the human makes only a few mistakes (n_h^{wrong}) because of following the robot. As P_{error} increases from 0.1 to 0.8, a slight increase can be seen in the number of wrong actions the human takes (n_h^{wrong}), the time that the robot needs to fix them (t_r^{wrong}), the number of tasks that the robot assigns to the human (n_r^{assign}), as well as the time (t_h^T and t_r^T), travel distance (d_h^T and d_r^T), and the total number of actions the human and robot needed to finish the tasks (n_h^T and n_r^T). Fig. 7a shows the cumulative number of assigned tasks by the robot to the human.

Moderate preference to follow the robot ($\mathbf{P_{follow} = 0.6}$): In this case, the robot has a moderate preference for following. The robot estimates the human's preference and, compared to $P_{follow} = 0.6$, assigns fewer tasks to the human. However, when the human's errors increase in number the robot assigns more tasks. Since the human does not entirely follow the robot, when P_{error} increases, n_h^{wrong} increases. With the increase of P_{error}, also n_r^{assign}, t_h^T and t_r^T, d_h^T, d_r^T, n_h^T, and n_r^T grow. The cumulative number of assigned tasks by the robot is depicted in Fig. 7b.

Slight preference to follow the robot ($\mathbf{P_{follow} = 0.3}$): When the human prefers to lead, n_r^{assign} shows that the robot can adapt itself to the human preference and assign fewer tasks. In this case, when the human is not accurate, the robot assigns significantly more tasks to retake the leading role and improve the team's performance. This case also shows an increase in n_r^{assign}, t_h^T and t_r^T, d_h^T, d_r^T, n_h^T, and n_r^T with the rise of P_e. Fig. 7c shows the cumulative number of assigned tasks by the robot to the human.

Travel distance - Total actions: This simulation assumes that the robot is slower than the human. Thus, to reduce collaboration time, the robot completes the tasks relating to workspaces 1 and 4 (W1 and W4) and leaves the tasks



(a) The human's errors (b) Assigned tasks by the robot

Fig. 8. Comparison of the results for the cases with adaptation (proposed algorithm) and without adaptation (ignoring the human's performance and preferences) when the human prefers to lead ($P_{follow} = 0.3$).

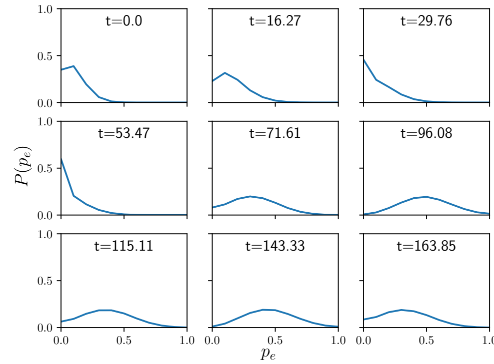
of workspaces 2 and 3 to the human as we assume that the human has a higher speed and is faster at doing them. Thus, when the human follows the robot, although their total working times are the same, the travel distance of the human is more than that of the robot. Additionally, the human has completed fewer tasks than of the robot has.

No adaptation ($\mathbf{p_e = 0}$, $\mathbf{p_f = 1}$): The value of n_r^{assign} shows that when the robot ignores the human preference for leading the robot, the robot will assign many tasks to the human, regardless of the human's accuracy level. Ignoring human preferences can lead to humans' distrust and dissatisfaction in real applications. More interestingly, although the robot assigns even more tasks than in the same cases in which it considers the human's preference and accuracy, the number of wrong actions by the human increases. These errors occur because the robot does not assign the right actions at the right time, allowing the human agent to make more mistakes. Fig. 8 depicts the number of the human's wrong actions and assigned tasks by the robot.

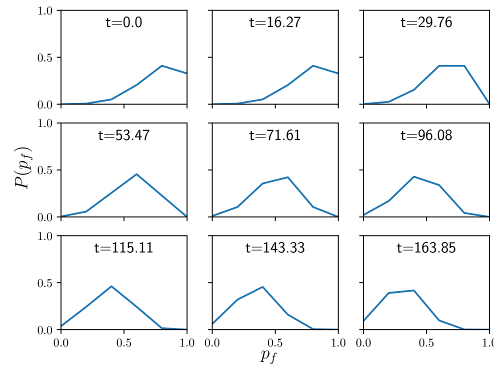
Probability distributions: For brevity, we show only the changes of probability distribution over the human's accuracy Fig. 9a and preference to follow Fig. 9b for the case ($P_{follow} = 0.3$, $P_{error} = 0.4$). As they show the robot can successfully estimate the human's preference and accuracy and get close to the values set for the human model.

V. CONCLUSIONS

We considered a task selection and planning problem in which the human and robot can select tasks and assign them to one another. We designed a planning architecture that adapt itself to the human's performance level and preference by updating its plan and its belief about the human online. The proposed algorithm was employed and evaluated in a simulated collaborative scenario. We considered a simple human model and applied the algorithm for different combinations of human accuracy and preference to lead. The results have shown that the proposed algorithm enables the robot to adjust its leading role based on the human's accuracy and preference. In addition, comparing our algorithm with the case that the robot only considers completion time shows a significant improvement in the team's performance.



(a) Robot belief about the human's accuracy



(b) Robot belief about the human's preference to follow

Fig. 9. probability distribution for the case $P_{follow} = 0.4$, $P_{error} = 0.5$

APPENDIX

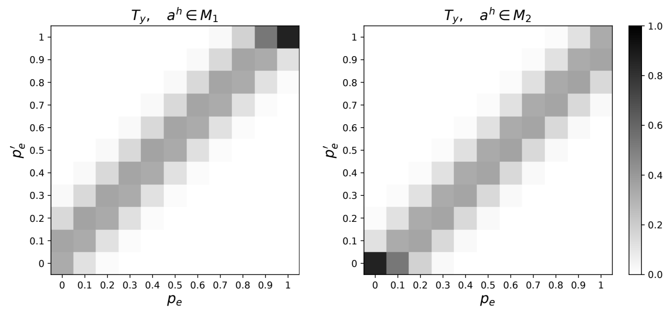
A. Updating p_e and p_f

The robot uses a history of k-step in human actions to model the human actions (policy). Factorizing the observable (X) and unobservable (Y) state variables of the system $S : X \times Y$, the belief update can be done as follows:

$$b'(y') = \eta Z(x', y', a^R, o) \sum_{y \in Y} T_x(x, y, a^R, a^H) \quad (10)$$
$$T_y(x, y, a^R, a^H, x', y') \pi^H(x, y, a^H) b(y),$$

where z is the observation function, T_x and T_y are the transition functions. π^H is the human action model (policy). **Following preference:** In this experiment, we assume that $Z = 1$ and $T_x = 1$. In addition, we assume that the human preference does not change or changes infrequently. Thus, we can write $T_y(x, y, a^R, a^H, x', y') = \delta_{y,y'}$, where $\delta_{y,y'}$ is the Kronecker delta function. For the human policy, we use a history of 3 steps in the human actions. The human's actions that the robot considers for updating P_f can be either assigning a subtask to the robot (F_1) and picking up (F_2) or do not picking up (F_3) an object from the tray when at least one is available. Letting f_1, f_2, and f_3 respectively the frequency counts of F_1, F_2, and F_3 in the set of k-step history of human actions, the human policy becomes:

$$\pi_f^H(x, y, a^H) = \begin{cases} \frac{\alpha f_1 + f_2}{\alpha f_1 + f_2 + f_3} y & a^H \in F_1 \cup F_2 \\ \frac{f_3}{\alpha f_1 + f_2 + f_3} y & a^H \in F_3 \end{cases}, \quad (11)$$



(a) Wrong action (b) Correct action

Fig. 10. Estimating the human's accuracy: Transition probability, T_y

where $\alpha > 1$ is a parameter to give more weight to the case when the human assigns a subtask to the robot.

Human error: We assume $Z = 1$ and $T_x = 1$. Modeling human error, and specifically, the humans' memory model in this scenario, is demanding and not the focus of this paper. However, we consider a simple model for T_y and π^H as they are required to estimate p_e. Defining $g_l(y)$ and $g_u(y)$ as the functions which return respectively the closest value less and closest value greater than y in set Y, we have

$$T_y = \begin{cases} p\big(y' \leq Z < g_u(y')\big), & \\ \qquad Z \sim \mathcal{SN}\big(g_u(y),\, \sigma^2, \beta_1\big) & \text{if } a^H \in M_1 \\ p\big(g_l(y') \leq Z < y'\big), & \\ \qquad Z \sim \mathcal{SN}\big(g_l(y),\, \sigma^2, \beta_2\big) & \text{if } a^H \in M_2 \end{cases}, \tag{12}$$

where $\mathcal{SN}(g_l(y),\, \sigma^2, \beta)$ is a skew-normal distribution function with the skewness factor β. The robot updates p_e based on the wrong actions (M_1) and correct ones if they are not assigned to the human by the robot (M_2). Fig. 10 show the transition probability T_y heatmap. Furthermore, the human's error model is

$$\pi_e^H(x, y, a^H) = \begin{cases} \frac{m_2}{m_1+m_2} y & a^H \in M_2 \\ \frac{m_1}{m_1+m_2} y & a^H \in M_1 \end{cases}, \tag{13}$$

where m_1 and m_2 are frequency counts of M_1, M_2

References

[1] F. Vicentini, "Collaborative robotics: a survey," *Journal of Mechanical Design*, vol. 143, no. 4, 2021.

[2] S. El Zaatari, M. Marei, W. Li, and Z. Usman, "Cobot programming for collaborative industrial tasks: An overview," *Robotics and Autonomous Systems*, vol. 116, pp. 162–180, 2019.

[3] A. Noormohammadi, A. Dahiya, A. M. Aroyo, S. L. Smith, and K. Dautenhahn, "The effect of robot decision making on human perception of a robot in a collaborative task-a remote study," in *Int. Conference on Human-Agent Interaction*, 2021, pp. 423–427.

[4] How robotic picking is revolutionizing warehouse productivity. [Online]. Available: https://www.dhl.com/global-en/delivered/digitalization/locus-robotics-robotic-picking.html

[5] A. Pasparakis, J. de Vries, and M. De Koster, "In control or under control? human-robot collaboration in warehouse order picking," *SSRN Journal*, 2021.

[6] L. Luo, N. Chakraborty, and K. Sycara, "Distributed algorithm design for multi-robot task assignment with deadlines for tasks," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3007–3013.

[7] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," *Cooperative robots and sensor networks 2015*, pp. 31–51, 2015.

[8] S. Choudhury, J. K. Gupta, M. J. Kochenderfer, D. Sadigh, and J. Bohg, "Dynamic multi-robot task allocation under uncertainty and temporal constraints," *Autonomous Robots*, vol. 46, no. 1, pp. 231–247, 2022.

[9] E. Lamon, A. De Franco, L. Peternel, and A. Ajoudani, "A capability-aware role allocation approach to industrial assembly tasks," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3378–3385, 2019.

[10] L. Johannsmeier and S. Haddadin, "A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 41–48, 2016.

[11] Y. Cheng, L. Sun, and M. Tomizuka, "Human-aware robot task planning based on a hierarchical task model," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1136–1143, 2021.

[12] A. Casalino, A. M. Zanchettin, L. Piroddi, and P. Rocco, "Optimal scheduling of human–robot collaborative assembly operations with time petri nets," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 1, pp. 70–84, 2019.

[13] F. Fusaro, E. Lamon, E. De Momi, and A. Ajoudani, "An integrated dynamic method for allocating roles and planning tasks for mixed human-robot teams," in *IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, 2021, pp. 534–539.

[14] S. Zhang, Y. Chen, J. Zhang, and Y. Jia, "Real-time adaptive assembly scheduling in human-multi-robot collaboration according to human capability," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3860–3866.

[15] A. Pupa, W. Van Dijk, and C. Secchi, "A human-centered dynamic scheduling architecture for collaborative application," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4736–4743, 2021.

[16] A. Tausch and A. Kluge, "The best task allocation process is to decide on one's own: effects of the allocation agent in human–robot interaction on perceived work characteristics and satisfaction," *Cognition, Technology & Work*, pp. 1–17, 2020.

[17] M. Gombolay, A. Bair, C. Huang, and J. Shah, "Computational design of mixed-initiative human–robot teaming that considers human factors: situational awareness, workload, and workflow preferences," *The International journal of robotics research*, vol. 36, no. 5-7, pp. 597–617, 2017.

[18] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan, "Learning from physical human corrections, one feature at a time," in *ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*, 2018, pp. 141–149.

[19] B. Akgun, M. Cakmak, K. Jiang, and A. L. Thomaz, "Keyframe-based learning from demonstration," *International Journal of Social Robotics*, vol. 4, no. 4, pp. 343–355, 2012.

[20] H. Soh, Y. Xie, M. Chen, and D. Hsu, "Multi-task trust transfer for human–robot interaction," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 233–249, 2020.

[21] C. Huang, W. Luo, and R. Liu, "Meta preference learning for fast user adaptation in human-supervisory multi-robot deployments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5851–5856.

[22] M. Chen, S. Nikolaidis, H. Soh, D. Hsu, and S. Srinivasa, "Planning with trust for human-robot collaboration," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 307–315.

[23] R. Luo, R. Hayne, and D. Berenson, "Unsupervised early prediction of human reaching for human–robot collaboration in shared workspaces," *Autonomous Robots*, vol. 42, no. 3, pp. 631–648, 2018.

[24] S. Nikolaidis, D. Hsu, and S. Srinivasa, "Human-robot mutual adaptation in collaborative tasks: Models and experiments," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 618–634, 2017.

[25] S. Nikolaidis, R. Ramakrishnan, K. Gu, and J. Shah, "Efficient model learning from joint-action demonstrations for human-robot collaborative tasks," in *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2015, pp. 189–196.

[26] V. V. Unhelkar and J. A. Shah, "Learning models of sequential decision-making with partial specification of agent behavior," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 2522–2530.

[27] M. C. Gombolay, R. J. Wilcox, and J. A. Shah, "Fast scheduling of robot teams performing tasks with temporospatial constraints," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 220–239, 2018.